Figure 1. KEY MAPPING

	Key	Visual Cue	Stroke/Radical	Examples Ues
	A	A	人亻𠂉日	会金大走奈谷幾仁气
	B	月	月月目	且肖肋且具
5	C	)	犭多豕魚馬龍鹿	
			牛羊虫鳥龜	豺狐豬家蒙
	D		丶广亠	方之太
	E	E ㅌE ヨヨヨ王	王隹Eㅌㄷヨヨヨ	門臣虐鳥龜書秉己尸弓改隻維
	F ヲ	𠃌ヲ	𠃌フ𠃌フマ土	方夕予刁過韋
10	G	ᓄ	辶辶臼	廸過健
	H	-	一工皿	正上
	I i	I i	讠丨言	中上正了丫丬小订
	J	J	丿	才广尸尹川用我
	K	ㄑ	丷冫巛ハ	丬四六八巛永小公冬头冲东東谷
15	L	L	乚レ乙厶	儿廿乩即宏能
	M	M	火灬	灶灰魚
	N n	𠂉冂	礻示冂	礼祭同過用舟而皿
	O	□	口	回呂嗎勛罵国因
	P	P	冂阝耳尸	部除服聆聳
20	Q	𠂤	女田	妝好
	r 卜	R卜	𠂉几飞	殳入之飛戈我
	S	𢆶	水氵灬弓	汁汞亟
	T t	十	十廿	千花子
	U	心	心忄	忙忘
25	V	✓	扌手	刁打挈
	W	W	竹癶癹	竺發登祭
	X	X	乂	攵文丈义 又夂
	Y	丰	木	村杏呆困本休米
	Z	礻	礻衣	良辰長派被衰
30				

Figure 2. SAMPLE CHARACTER SEQUENCES

1. 躬 - This complex character is composed of simpler strokes and/or characters:

丿 丨 ㇆ 一 一 一 丿丿 ㇆ 丷 木

j i f h h h j j f k y (key encoding)

Therefore, the encoding is: "jifhhhjjfky". However, 身 is also a character represented by the encoding "jifhhhj", which is abbreviable to the encoding "j". So the encoding "jjfky" will also identify the character 躬.

Furthermore, 米 is also a character and abbreviable to the encoding "y". Therefore, the encodings "jifhhhjjfy" and "jjfy" identify the character 躬.

2. 叶 - This character is composed of two other characters 口 (means mouth) and 十 (means the number 10):

口十

o t (key encoding)

Therefore, the encoding is: "ot"

3. 古 - This character is composed of the same two characters as the above character but in a different order:

十口

t o (key encoding)

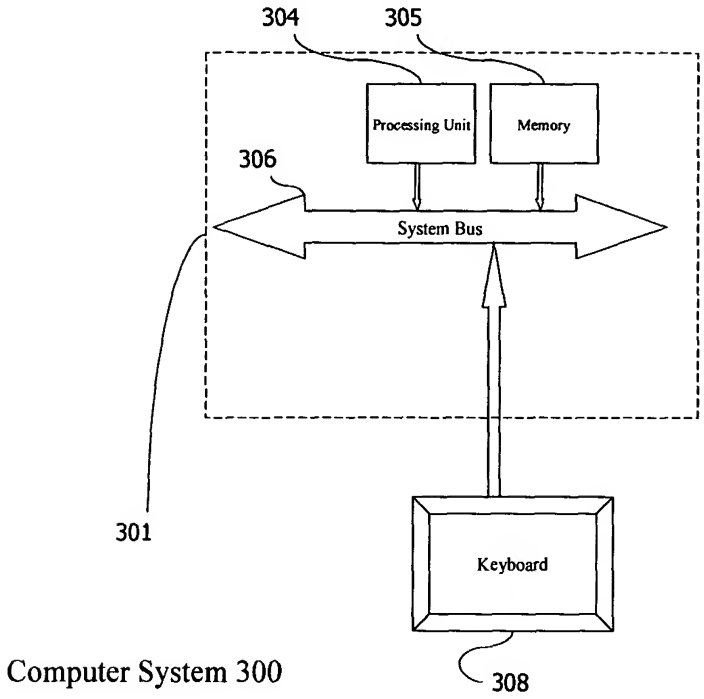
Therefore, the encoding is: "to"

4. 困 - This character is made up of the character 口 and 木. So the encoding should be "oy". However, to distinguish it from the following

character, which is also composed of the same component characters, a positional encoding is added, making the encoding “oyx”.

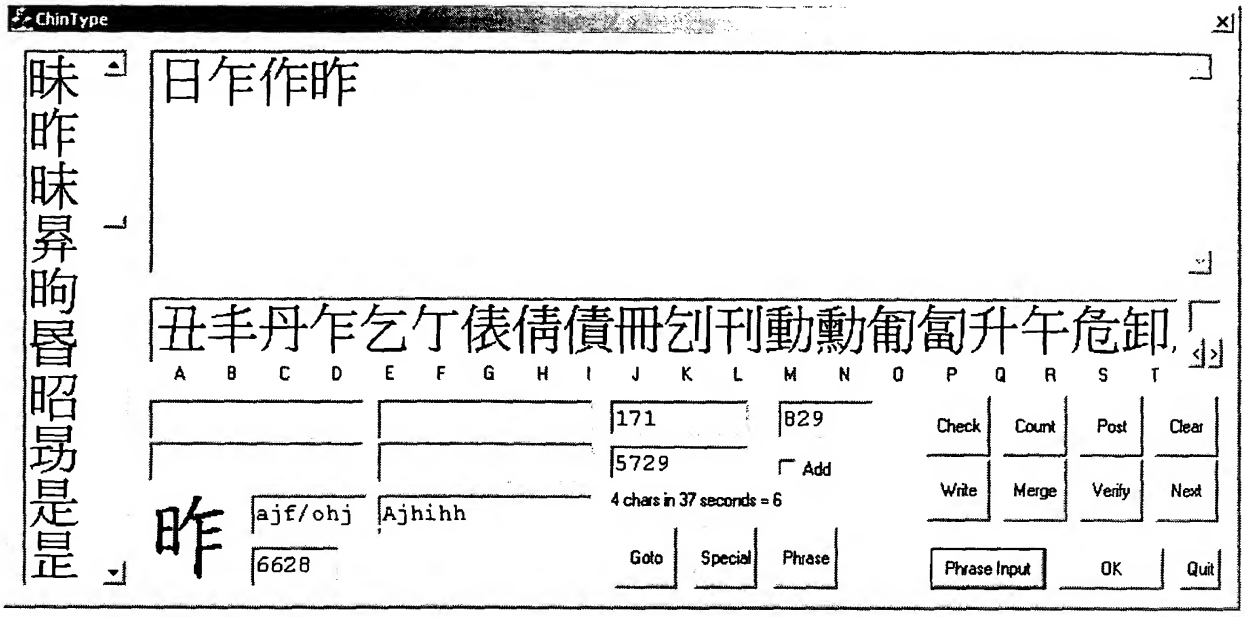
5. 呆- This character is also made up of 口and木, so the encoding should also be “oy”. To create a distinct encoding, a positional encoding is added, making the encoding “oyz”.

Figure 3. SYSTEM DIAGRAM



Computer System 300

Figure 4. USER VIEW OF ONE EMBODIMENT



In this particular embodiment of the present invention, a computer program running on personal computer interacts with a user to receive input text strings and then comparing them to the predefined strings stored in the program is used. The large top window is the output window and shows Chinese characters that have been successfully identified and accepted. The long flat window below is the candidate window which shows characters that have passed certain selection criteria given an input text string and awaiting manual selection by the user pressing a letter between 'a' to 't' corresponding to the label of the candidate below each candidate.

Figure 5. AN EXAMPLE BACKWARD EQUIVALENCE TABLE

Below is a code fragment in C language defining one version of a "backward equivalence table" as described above. Each letter in the encoding language's alphabet may be defined to be equivalent to 0 or more encodings.

```
radDef dumbrads[22] = { 2, L"oh",  // a - 0
                        4, L"ifhh", 4, L"jfhh",        // b - 1, 2
                        // c
                        // d
                        8, L"jidhhhih", 7, L"adhhhih", 3, L"hth", 3, L"hlh", 3,
L"fhh", 2, L"hl", 2, L"fh",// e - 3..9
                        // f
                        // g
                        // h
                        5, L"dhhho", // i
                        // j
                        2, L"dd", // k - 10
                        // l
                        4, L"dddd",  // m - 11
                        4, L"hhik", // n - 12
                        // o
                        6, L"hiihhh", 3, L"ffi", 2, L"fi",// p - 13, 14, 15
                        4, L"ahih", // q
                        // r
                        3, L"ddd", // s - 32, 33, 34
                        2, L"hi",   // t - 35..38
                        // u
                        4, L"jhhi",// v - 44, 45, 46
                        // w
                        // x
                        // y
                        // z
                      };
```

Figure 6. AN EXAMPLE FORWARD EQUIVALENCE TABLE

Similar in structure to the backward equivalence table but used to expand input strings instead of predefined strings during a string comparison.

```
radDef smartRads[238] = { 3, L"jha", 2, L"ji", 2, L"jh", 1, L"A",  // a - 0, 1, 2, 3
                          7, L"ifhidhb", 6, L"hjohhk", 6, L"ifhhhh", 5, L"ohhjl", 5, L"ifhhh", 5,
                             L"jifkh", 4, L"ifhh", 4, L"ifkh", 3, L"ohh", 2, L"ok", 1, L"K", // b
                         12, L"dhkhbihelhhh", 11, L"jffhiexllee", 10, L"dhjeiilhlh", 7, L"ihhjhlr",
                             7, L"hhhilfm", 6, L"jeihfm", 6, L"ihhjhl", 5, L"jfotm", 5, L"khhhi", 4,
                             L"jhhi", 4, L"oihd", 3, L"had", 3, L"hli",   // c - 14..25
                          7, L"taaajfd", 7, L"taaajfr", 6, L"jjihdr", 5, L"hldth", 5, L"ifihd", 4,
                             L"hlfd", 4, L"hjfd", 4, L"ddhk", 3, L"jfd", 3, L"ddh", 3, L"dhj", 2,
                             L"td", 2, L"id", 2, L"ld", 2, L"fd", 2, L"dh", 2, L"dh", // d - 26..42
                          5, L"adeld", 5, L"adeld", 4, L"eiei", 4, L"eioi", 3, L"aek", 3, L"eil", 2,
                             L"el", 2, L"ej", 2, L"ed", // e - 43..49
                          8, L"fihohifi", 7, L"fihohli", 6, L"thjlfh", 5, L"thjlh", 5, L"thiha", 5,
                             L"oihih", 4, L"oiha", 4, L"hiha", 4, L"ihih", 4, L"dhjf", 4, L"jhhh",
                             4, L"ihhh", 3, L"ihh", 3, L"iha", 3, L"thj", 2, L"ih", 2, L"th", 2,
                             L"jf", 2, L"fj",  // f - 50..67
                         11, L"jifhhhlklkr", 7, L"jifhhhi", 6, L"jifhhh",          // g - 68..70
                          4, L"jhhi", 3, L"hih", 2, L"hj", // h - 71..73
                          3, L"elf", 1, L"I", // i - 74, 75
                         10, L"hhhihldjjj", 7, L"jifhhhj", 5, L"fdhij", 5, L"jihhi", 4, L"hlij", 4,
                             L"lfhj", 4, L"jjhi", 4, L"johh", 3, L"jji", 3, L"jjj", 3, L"hjo", 3,
                             L"jto", 3, L"jyA", 3, L"jsy", 2, L"jA", 2, L"fj", 2, L"jf", 2, L"hz",
                             // j - 76..93
                          6, L"jyaikk", 6, L"khjohh", 6, L"hifikk", 5, L"dhkhA", 5, L"dhkht", 5,
                             L"thjik", 5, L"jjfhk", 4, L"lfhk", 4, L"dhkh", 4, L"fkfk", 4, L"eikk",
                             4, L"hokh", 4, L"khha", 4, L"dhjk", 3, L"kao", 3, L"kkk", 2, L"fk", 2,
                             L"ky", 2, L"kt", 2, L"ik", // k - 94..113
                          5, L"jfeil", 4, L"oeil", 4, L"jhhl", 4, L"lhlh", 3, L"eil", 2, L"el", 2,
                             L"fl", 2, L"jl", // l - 114..121
                         12, L"jihhifhofhhh", 12, L"jihhihfifhhh", 11, L"jihhxxfhhdh", 7, L"kifjjlr",
                             6, L"okhihm", 5, L"ihoxm", 4, L"lixm", 4, L"yxxy", 3, L"oxm",  // m -
                             122..130
                         10, L"ahoooifiih", 7, L"hoifkhi", 6, L"dhoifo", 6, L"jfjfth", 6, L"hjifii",
                             5, L"jfhhi", 5, L"ifiih", 4, L"ifaa", 4, L"hhik", 3, L"ifa", 3, L"ifi",
                             2, L"if", 2, L"jf", // n
                          4, L"hjoo", // o
                          5, L"jieet", 4, L"jiee", 3, L"dej", 3, L"eij", 2, L"ej", 2, L"ei", // p -
                             144..149
                          9, L"johhothji", 7, L"hlihAik", 7, L"jotjlld", 4, L"Ahih", 2, L"ot", 2,
                             L"ei", 2, L"Ai", 1, L"Q", // q
                          4, L"jhhf", 4, L"hrjd", 4, L"jjir", 3, L"hrd", 3, L"rjd", 3, L"hjr", 3,
                             L"jfl", 2, L"jl", // r - 156..162
                          4, L"dffk", 3, L"elf", 3, L"ikk", // s - 162..163
```

```
    6, L"hiihhh", 5, L"hhihb", 5, L"hliot", 5, L"jhhih", 5, L"hhiih", 4,
        L"ehhi", 4, L"ehih", 4, L"hiih", 4, L"hhih", 4, L"jhhi", 4, L"hhhi", 3,
        L"hhi", 3, L"hji", 2, L"ft", 2, L"ht", 2, L"hi", 2, L"tt", 1, L"T",
        // t - 164..179
    11, L"ihihaahaali", 5, L"jhtli", 4, L"hlih", 3, L"hli", 3, L"lii", 2, L"li",
        // u - 180..185
    // v - 186
    8, L"ihlilihi", 8, L"jhhhihhh", 6, L"jfdhli", 5, L"lldik", 5, L"dhlld", 5,
        L"jldfj", 5, L"jihhi", 4, L"lihj", 4, L"llds", 3, L"lld", 3, L"llj", 2,
        L"pq", 1, L"W", // w - 186..197
    7, L"thokhtx", 4, L"hijx", 3, L"dhx", 3, L"ihx", 3, L"jhx", 3, L"hjo", 2,
        L"jx", 2, L"hz", 2, L"kx", 2, L"rx", 2, L"xx", // x - 198..207
    5, L"dhjyy", 5, L"hhhik", 4, L"lihj", 4, L"jhha", 2, L"ha", 2, L"ky", 2,
        L"jy", 2, L"li", 2, L"yd",    // y - 208..216
    5, L"hhhiz", 5, L"hhhjz", 5, L"jlhrh", 5, L"jlhrd", 4, L"jlhr", 2, L"ez"
        // z - 217..222
};
```